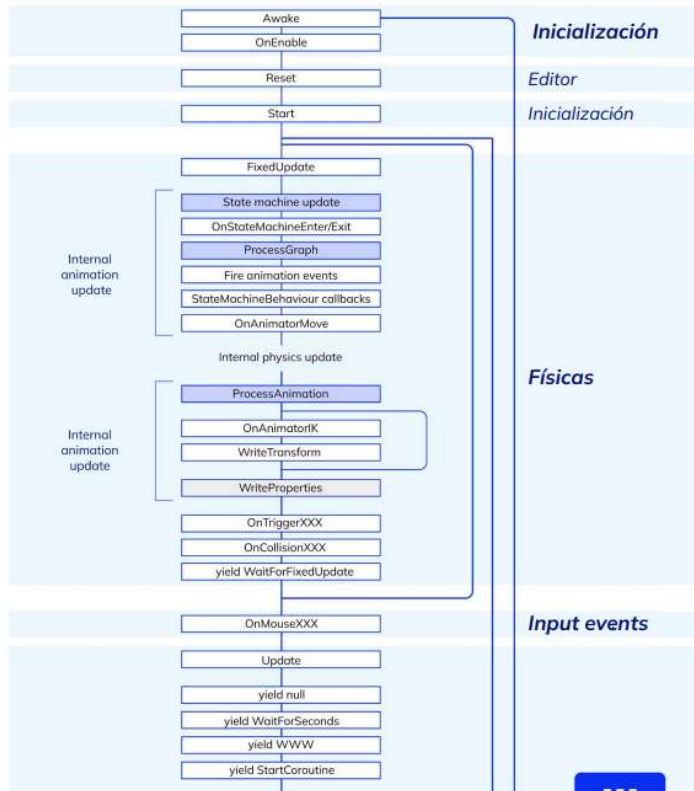


Game Loop

Todo en Unity tiene un orden. EL orden es el siguiente:

- 1) Inicialización
- 2) Físicas
- 3) Inputs
- 4) Game Logic (Unity / Logica de programación)
- 5) Rendering

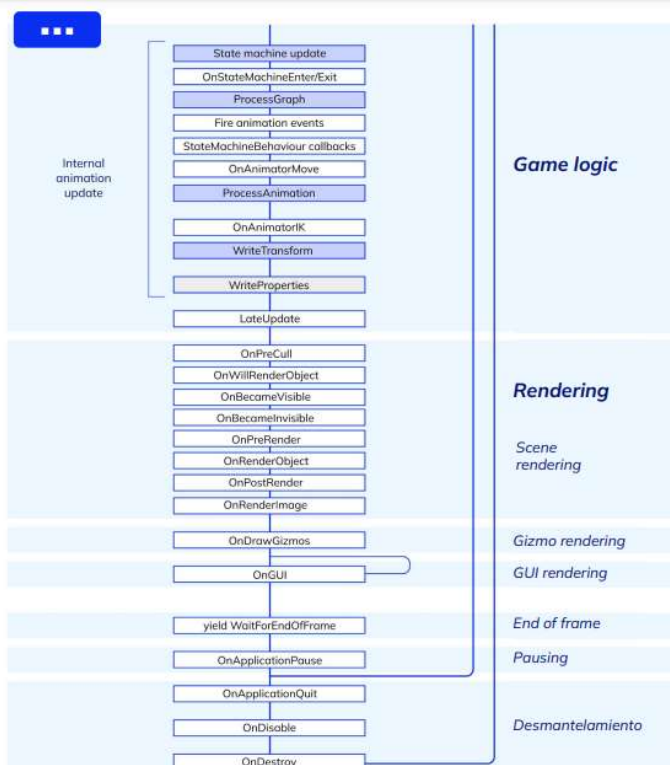
Ciclo de vida del script



Referencias

User callback
Internal Multithreaded function
Internal function

(Continúa en la diapositiva siguiente)



Escena: Es un pedazo del juego. Donde podemos mostrar diferentes elementos.
Se pueden crear directamente en los assets.

Scripts

Componentes que agregar a un Game Object. Se utiliza C# como lenguaje para la codificación.
Rider es mejor que Visual Studio. Create un script es dentro de assets.
Plantilla por default en unity.

Using: Librerías a utilizar.

Una clase por script. Darle logica al elemento que estamos generando.

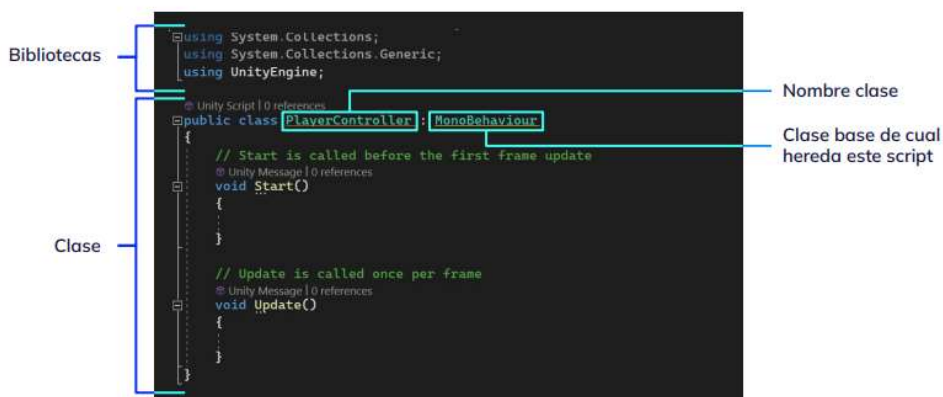
Public class NewBehaviourScript : MonoBehaviour

Scripts

Ciclo de vida de los métodos de los scripts de Unity	
Awake()	Este evento se llama una vez cuando se crea un objeto con un script adjunto en la escena. Es útil para inicializar variables y configurar componentes antes de que comience la ejecución del juego.
OnEnable()	Este evento se llama cuando el objeto se activa. Es útil para realizar tareas que solo deben realizarse cuando el objeto está activo.
Start()	Este evento se llama una vez al inicio del juego, después de Awake() . Es útil para inicializar variables y configurar componentes que dependen de otros objetos.
FixedUpdate()	Este evento se llama en un intervalo de tiempo fijo y se utiliza para actualizar la física del juego. Es útil para realizar cálculos que requieren una precisión constante.
Update()	Este evento se llama una vez por frame y se utiliza para actualizar la lógica del juego. Es útil para realizar tareas como mover objetos, actualizar la cámara y procesar la entrada del usuario.
LateUpdate()	Este evento se llama una vez por frame después de que se hayan procesado todos los Update() en los objetos de la escena. Es útil para actualizar la cámara y otros objetos que dependen del movimiento de otros objetos.

OnGUI()	Este evento se llama una vez por frame después de que se hayan procesado todos los Update() y LateUpdate() en los objetos de la escena. Es útil para mostrar elementos de interfaz de usuario en la pantalla.
OnDisable()	Este evento se llama cuando el objeto se desactiva. Es útil para limpiar o liberar recursos cuando el objeto ya no está activo.
OnDestroy()	Este evento se llama cuando el objeto se destruye. Es útil para limpiar o liberar recursos cuando el objeto ya no es necesario.

Estructura de un Script



Void Update se ejecuta a cada frame.

Sistemas Físicas

Ejercicio: Generar que se aplique la gravedad a una esfera al momento de tocar la tecla space.

El metodo se debe llamar al update porque suceden todos los frames. Todos los frames pregunto si se presiona esa tecla.

Nota: Renombrar variables : control + r + r

Find and replace : control + h

Revisar que hace el GetComponent.

En Unity, el método GetComponent se utiliza para acceder a un componente adjunto a un GameObject. Los GameObjects en Unity pueden tener varios componentes adjuntos, como Transform, Rigidbody, Collider, Script, etc. GetComponent permite acceder a uno de estos componentes para interactuar con él o para acceder a sus propiedades y métodos.

Por ejemplo, si tienes un GameObject con un componente Rigidbody adjunto y deseas acceder a ese componente en un script para manipular la física del GameObject, puedes usar GetComponent<Rigidbody>() para obtener una referencia al componente Rigidbody. Luego puedes acceder a las propiedades y métodos de ese componente, como velocity, mass, AddForce, entre otros.

Escena nueva.

Pelota que cuando toco barra espaciadora empieza a caer pero quiero que me digan cuanto tiempo está cayendo.

Después otra pelota, que todo lo que toque lo destruya.

Cuantos frames y despues se pasa a tiempo.

Update: Cada frame le sumo uno.

Metodo de colision: OnCollisionEnter()

Time.deltaTime(herramienta de unity)